Illustrating Terrains using Direction of Slope and Lighting

Kevin Buchin[1] Mario Costa Sousa[2] Jürgen Döllner[3] Faramarz Samavati[2] Maike Walther[1]

[1]Institute of Computer Science, Free University of Berlin, {buchin,mbuchin}@inf.fu-berlin.de

[2]Department of Computer Science, University of Calgary, {mario,samavati}@cpsc.ucalgary.ca

[3]Hasso-Plattner Institute, University of Potsdam, doellner@hpi.uni-potsdam.de

Abstract

Landscape illustrations and cartographic maps depict terrain surface in a qualitatively effective way. In this paper, we present a framework for line drawing techniques for automatically reproducing traditional illustrations of terrain by means of slope lines and tonal variations. Given a digital elevation model, surface measures are computed and slope lines of the terrain are hierarchically traced and stored. At run-time slope lines are rendered by stylized procedural and texture-based strokes. The stroke density of the final image is determined according to the light intensities. Using a texture based approach, the line drawing pipeline is encapsulated from the rendering of the terrain geometry. Our system operates on terrain data at interactive rates while maintaining frame-to-frame coherence.

Key words: Terrain visualization, non-photorealistic rendering.

1 Introduction

For realistic landscape depiction a vast amount of detail is desirable. In contrast, the effective and efficient communication of spatial information requires a careful choice and an adequate form of representation. A heuristic approach to this task is the algorithmic realization of traditional drawing techniques, which is based on the assumption that these techniques have been optimized over time. Techniques for terrain drawing can be found in architectural landscape illustration and cartography (fig. 1). Traditionally, the aim of cartographic terrain depiction is the two-dimensional representation of three-dimensional terrain in a geometric precise, illustrative, and comprehensible manner. The goal of an artistic landscape drawing is not necessarily the exact representation of the terrain, nevertheless an artistic drawing can still be of high topographic quality. The sacrifice of precision might well be compensated by a more vivid impression of the landscape.

In this paper we present results from our research in terrain illustration methods using Non-Photorealistic Rendering (NPR). The main motivation for this work is to investigate the computer-generated reproduction of traditional terrain illustration techniques to provide alternate display models for geoscientists, cartographers, and architects. We chose to reproduce two particular techniques: loose lines and hachures illustrations as shown in Figure 1. Loose lines suggest the direction of terrain slope by means of continuous lines with varying thickness and perturbation. Hachures are also lines drawn in the direction of slope but without perturbation and with thickness corresponding to slope steepness or light intensity. In traditional hachure maps they are interrupted at contour lines. We omit this since in 3D terrain depiction contour lines as quantitative measure of height are not necessary.

The main contribution of our research is on the modeling and implementation of an integrated framework for terrain line rendering based on quantitative surface analysis and NPR algorithms. Our approach was to design a system which allows the reproduction of different line styles and techniques. We demonstrate this aspect by reproducing the two traditional techniques of loose and hachure lines.

2 Related work

The investigation of computer-generated terrain illustration techniques is receiving increased attention from the NPR community as well as from geoscientists, cartographers, and landscape architects.

Early experiments in computer-aided line drawings of terrain include the work of Yoeli [25] who created large-scale slope and shadow hachures using a plotter. In the context of landscape architecture, Sasada [17] demonstrated the use of simple, thin, single-width line marks indicating ridges of terrain data observed from a distance.

Important results in algorithmic sketching of terrain have come from the Cartographic Information Systems Research Group (CISRG) of the University of Hull. They introduced P-strokes [2, 20] used for sketching filtered portions of terrain profiles orthogonal to the camera, occluding contours [21] for enhancing P-stroke sketches, edge detectors for extracting terrain sketching lines [12]

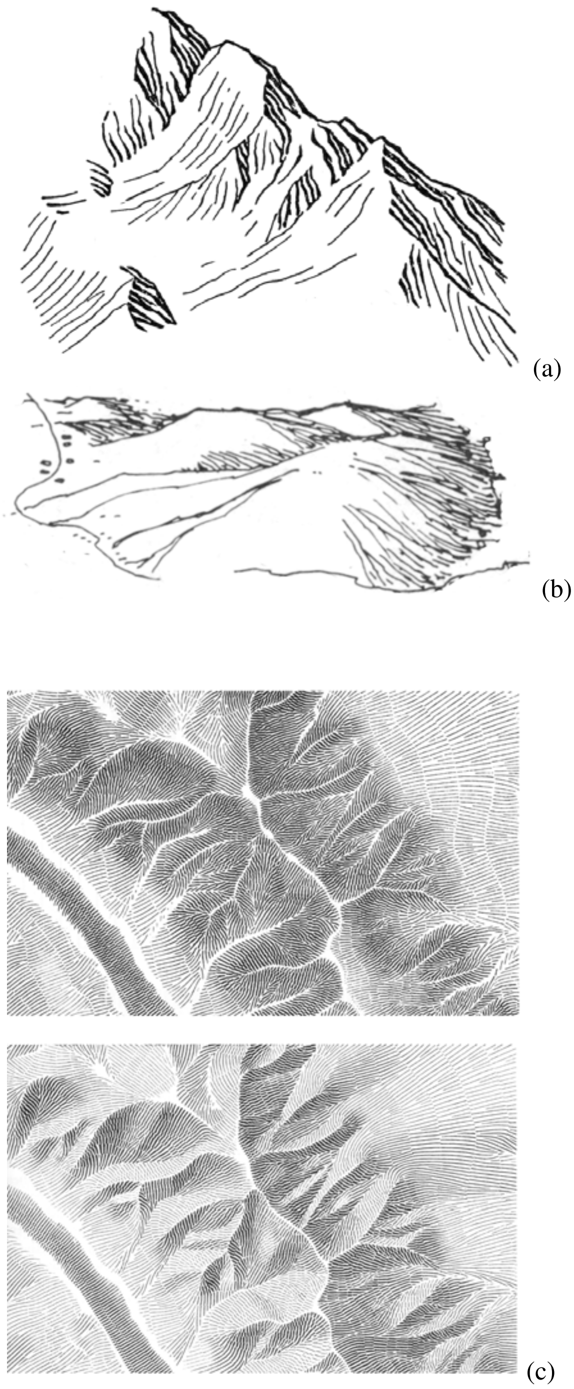


Figure 1: Traditional ink drawings of terrain using loose lines for (a) cartography [7], (b) landscape architecture [3], and (c) using hachure style for cartography, with slope steepness (top map) and lighting (bottom map) as thickness parameter [8].

and more recently a slope filter to abstract visible slopes with negative gradients [23].

A related field of research is the development of computer tools for aiding the process of map making. This field is also concerned with the simulation of traditional cartographic techniques but for 2D maps and not 3D terrain depiction. Hurni et al. [5] presented a system for semi-automatic generation of cliff drawings by ridge, contour and drain lines. Hurni et al. [6] introduced a semi-automatic system and methods for digital relief depiction using fill hachures. Kennelly and Kimerling [10] described a method for rendering hachures with strokes as arrowed lines following the direction of slope, rendered as white or black if front or back facing the light source, respectively.

3 System overview

Typical elements of a digital terrain modeling system are techniques for the generation, manipulation, interpretation and visualization of a digital terrain model. Information about the terrain surface can be extracted through interpretation by quantitative surface analysis, and through visualization by visual analysis [22]. Our system is designed for supporting and integrating quantitative surface analysis and visualization by means of a NPR line-based illustration pipeline.

One important aspect of our system is that the rendering pipeline is independent of the geometry pipeline. We use a texture-based approach for drawing the lines onto the terrain geometry, i.e. the lines are rendered into an offscreen-buffer and then projected onto the geometry as a texture. This texture is composed during the rendering pipeline, stored in an offscreen buffer and later projected to the terrain geometry. The architecture of our system is presented in Figure 2. Boxes and ellipses correspond to objects and actions, respectively.

Given a terrain data (elevation matrix), our system initially integrates it with a 3D multiresolution model, which represents the terrain geometry for rendering [1, 13]. Next, our system performs two pre-processing steps: (1) quantitative surface analysis by locally approximating surface measures (subsec. 3.1) and (2) path computation of every stroke following slope (elevation gradient) direction (subsec. 3.2). At run-time, a subset of slope lines (stroke paths) is selected based on lighting information (subsec. 3.3). Stroke qualities of thickness and perturbation are then procedurally defined for each path selected for display (subsec. 3.4). Thickness variation is a function of surface measures and perturbation is adjusted according to waviness-based functions to suggest small hand-gesture variations. In our system, texture maps of real hand-drawn strokes can also be assigned to the fi-

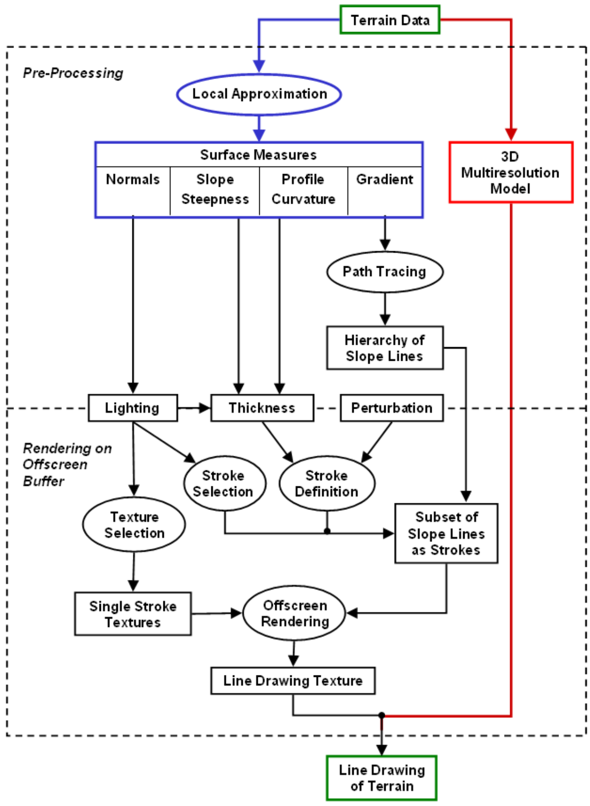


Figure 2: System pipelines: geometry (red), measurement (blue) and rendering (black).

nal strokes (subsec 3.4). The selected strokes for the final image are projected onto the terrain geometry using texturing mechanisms for either raster data [1] or vector data [11]. Conceptually, this is done by rendering the strokes into an offscreen buffer and projecting the content of this buffer onto the terrain geometry.

Figure 3: Stroke placement following direction of slope. From left to right: given a terrain data, the first layer with 4 strokes and progressing up to the last layer with 79 strokes.

This system framework allows for general line drawings, making it flexible to reproduce traditional styles. We demonstrate it by reproducing loose lines and hachures (subsec. 3.5).

3.1 Quantitative surface analysis

We assume the terrain data is given as an elevation matrix DEM (i.e. a uniformly sampled height field). For each point $\mathsf{h(x,y)}$, we locally approximate the height field by a quadratic function resulting in first and second derivatives, denoted respectively by

$$(\mathrm{p},\mathrm{q})=\left(\frac{\partial h}{\partial x},\frac{\partial h}{\partial y}\right)(x_0,y_0)$$

$$(\mathrm{r},\mathrm{s},\mathrm{t})=\left(\frac{\partial^2 h}{\partial x^2},\frac{\partial^2 h}{\partial xy},\frac{\partial^2 h}{\partial y^2}\right)(x_0,y_0)$$

Different map scales can be achieved by choosing different neighborhood sizes for the local approximation (see fig. 10).

Next, we compute *morphometric variables* (MV), a set of numerical values given per sample point [18]. MVs describe general geomorphometric properties of a terrain surface, and typically correspond to groups of 0, 1st and 2nd order differentials. Evans [4] considers the MVs elevation (0 order), slope steepness and aspect (1st order) and profile and plan curvature (2nd order). For the calculations in this paper we use the following measures:

Gradient of the height field $\mathsf{h(x,y)}$:

$$\mathsf{G} = (\mathsf{p},\mathsf{q}) \tag{1}$$

Note that this is different from the geomorphological term *slope gradient* which refers to the length of G.

Slope steepness showing the rate of change in elevation:

$$\mathsf{GA} = \arctan(\mathsf{p}^2+\mathsf{q}^2)^{0.5} \tag{2}$$

Profile curvature, the curvature of the corresponding normal section, which is tangential to a slope line.

$$\mathsf{kv} = \frac{-(\mathsf{p}^2\mathsf{r}+2\mathsf{pqs}+\mathsf{q}^2\mathsf{t})}{(\mathsf{p}^2+\mathsf{q}^2)(1+\mathsf{p}^2+\mathsf{q}^2)^{1.5}} \tag{3}$$

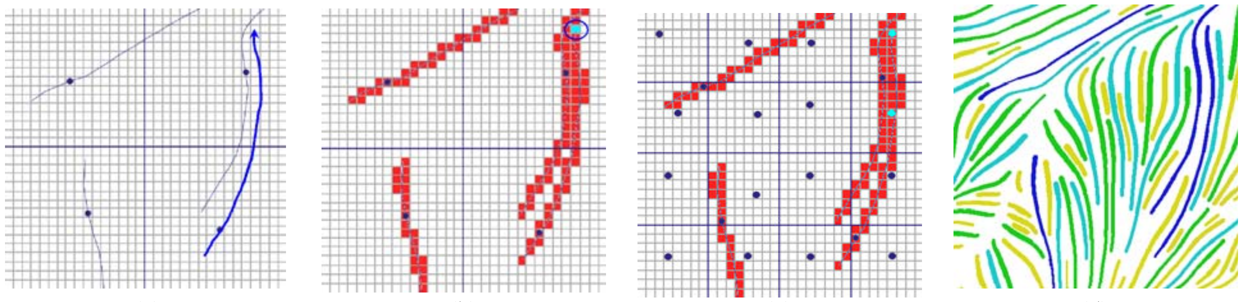
Positive profile curvature indicates convex and negative concave profiles.

Normal,

$$\mathsf{N} = (-\mathsf{p},-\mathsf{q},1)/(1+\mathsf{p}^2+\mathsf{q}^2)^{0.5} \tag{4}$$

3.2 Hierarchical path tracing

Our goal now is to cover the whole terrain data with sets of non-overlapping strokes following the direction of slope (fig. 3). In our system all the stroke locations and paths are determined in a pre-processing stage. As a first step, a reference image I is created to record where each stroke will be placed and drawn (and therefore, no further stroke will be drawn). Our system then iteratively computes a hierarchy of stroke lines to allow selective control



(a) (b) (c) (d)

Figure 4: Example of the path tracing and hierarchy building process for the terrain data in Figure 3. (a) First layer with four stroke seed points and corresponding paths (stroke in light blue is the last being traced). (b) Reference image $\boldsymbol{I}$ with corresponding updates of path tracing in (a). The blue path from (a) ends because it hits a position in $\boldsymbol{I}(x, y)$ which is already marked (note the encircled cyan pixel). (c) Seed points for the next subdivision layer. (d) Possible final stroke hierarchy with 79 strokes distributed in 4 layers. Stroke colors correspond to layer numbers: # 1 (blue, 4 strokes), #2 (cyan, 13 strokes), #3 (green, 23 strokes), #4 (yellow, 39 strokes).

over stroke densities as a function of lighting information (subsec. 3.3).

Refer to Figure 4. We start off with a coarse subdivision of I. The midpoint of each section corresponds to a seed point indicating the placement location for the stroke (fig. 4a). In each step, our system computes one level of the hierarchy as follows: (1) a stroke path is determined for each seed point (fig. 4a); (2) the reference image I is updated with this new stroke path (fig. 4b); (3) the stroke line is stored in a list of strokes associated with the current hierarchy level, and (4) the subdivision is refined by dividing each of the sections in four new sections (fig. 4c). Iteration ends when the number of strokes for one level drops below some threshold (fig. 4d).

Finding slope lines

Essentially, the problem of finding and tracing slope lines is a problem of 2D vector field visualization [14]. In our system, we let $x(t)$ be the stroke path around the given point P. We want to find $x(t)$ such that when we move along it, the direction of the movement, $\dot{x}(t)$, will be the same as the direction of the steepest slope of the height field. The gradient vector of height field gives the steepest slope direction, consequently, $x(t)$ should satisfy the following differential equation: $\dot{x} = \pm \mathrm{grad}\ h(x(t))$, with initial condition $x(0) = P$. Note that we locally approximated the heightfield by a quadratic function and consequently, we have the gradient function. Using the Runge-Kutta method, we solve this first order differential equation with the initial condition.

Keeping track of slope lines

We also control the spacing between two strokes so that they do not overlap. In principle, our approach is similar to streamline placement techniques as presented by Jobard and Lefer [9] but we choose the cell size differently to avoid distance computations within a cell. In our system, the idea is that we keep track of the lines already traced by drawing strokes into a reference image I (with a bitmap-brush approximating a circle). This reference image has size: $(DEM_{size}\ brush_{size})/(2dist_{min})$, where DEM_{size} denotes the size of the digital elevation model in x and y coordinates, $dist_{min}$ denotes the minimal distance allowed between two stroke paths (in the same coordinate system as DEM_{size}) and $brush_{size}$ denotes the size of the bitmap representing the brush.

While tracing a stroke we check in the reference image whether there is an intersection with a stroke that has already been drawn (and recorded in the reference image). After tracing, if the stroke is accepted (i.e. the length is above some threshold), it is drawn into the reference image. For the intersection tests, $dist_{min}$ needs to be at least the maximal width of a stroke.

3.3 Selecting strokes by light

In our system, the stroke density of the final image is determined by selecting strokes from the stroke hierarchy according to the light intensities and thresholds of the layers of the hierarchy. The idea is that strokes from higher and lower layers will be selected for higher and lower light intensities, respectively (fig. 5). Because the strokes in high layers of the hierarchy are traced first, they are longer than those in lower layers and depict the overall structure of the terrain, while strokes in lower layers are shorter and fill areas of the terrain. More details on the selection of strokes from the hierarchy are given next.
Every layer i in the hierarchy has a threshold value t_i associated with it. For layers i and j, $i < j$, the corre-

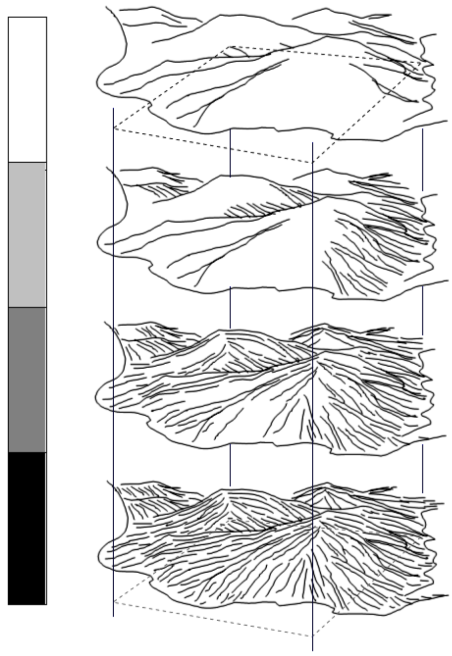


Figure 5: Correspondence between tone values and stroke densities on a terrain.

sponding threshold values t_i and t_j fulfill $t_i \geq t_j$. Our system chooses values such that the (t_i, t_j) give intervals of equal length between estimated minimal and maximal light intensity (this can be estimated using the slope of the light and the maximal slope of the terrain). The user has the possibility to change these values for an optimal rendering. A particular stroke of a layer i is selected for display, if $N \cdot L < t_i$. Here, N is the normal associated with every stroke, averaged during the path tracing over the chain point normals. L is the light direction, typically positioned with a vertical angle of 30 to 45 degrees.

To give an example of how threshold values t_i are determined, let us assume that the terrain has a maximal slope steepness of 30 degrees, and a vertical light angle of 30 degrees. Then $N \cdot L$ might vary between

$$\text{light}_{max} = \cos(30^o) = 0.866$$
$$\text{light}_{min} = \cos(90^o) = 0$$

If we have layers $1, ..., n$ we define

$$\lambda = (\text{light}_{max} - \text{light}_{min})/(n + 1)$$

and for layers 1 to n choose:

$$t_i = \text{light}_{max} - \lambda i$$

Note that light_{max} and light_{min} can be adjusted by the user as well. Also, light_{min} is allowed to be negative.

3.4 Rendering individual strokes

At this stage, we include attributes along the stroke paths to approximate the visual qualities of traditional pen-and-ink strokes. Our system supports three main approaches, that can be used independently or in combination: procedural thickness control, perturbation and mapping of individual tone-valued stroke textures. These attributes are then adjusted to approximate the reproduction of specific traditional terrain illustration techniques.

Procedural thickness control

Refers to the simulation of the traditional effect of placing different amounts of ink along the stroke. In our system, stroke thickness is controlled as follows: For every chain point $P = (x_0, y_0)$ in the path, evaluate V, the corresponding surface measure (slope steepness or profile curvature) or light intensity. The user can specify a stroke thickness between $[W_{min}, W_{max}]$, and the corresponding MV or light values $[V_{min}, V_{max}]$. A quadstrip extrusion in two-dimensions is then computed in both direction $(+, -)$ corresponding to (left, right) to P, respectively: The new extruded points P^0 are calculated as:

$$P^0 = P \pm 0.5\, N_g(V W) \tag{5}$$

where,

$$V = (V - V_{min})/(V_{max} - V_{min}), \in [V_{min}, V_{max}],$$
$$W = (W_{max} - W_{min}) + W_{min},$$

and N_g denotes the normalized vector orthogonal to the gradient $G = (p, q)$ (subsec. 3.1) at P, computed as:

$$N_g = (-q, p)/(p^2 + q^2)^{0.5} \tag{6}$$

The product by 0.5 in equation 5 forces the intended width to go half into the direction of N_g and the other half into the direction of $-N_g$.

Procedural path perturbation

Refers to the simulation of small hand-gestures suggesting “artistic” irregularities in the drawing. We adapted the function $PerturbedLineSegment()$, introduced by Salisbury et al [16].

Tonal stroke textures

The goal here is to use real hand-drawn pen-and-ink strokes used in loose drawing style for terrains. Each stroke corresponds to a particular tone value. We scan the strokes and organize them in a look-up table indexed by intensity values, as shown in Table 1. This approach has been used previously to store and index collection of strokes per tone [15, 19, 24]. In our case we store single strokes per tone value range.

3.5 Reproducing traditional techniques

Different traditional line drawing techniques can be approximated with our terrain NPR line drawing pipeline. We experimented with loose and hachure lines (fig. 1). Loose lines can be considered the most general reproduction. All steps of the pipeline are used. For hachures, three constraints are necessary: (1) no perturbation, (2) no stroke selection through lighting and (3) no single stroke textures. For hachures, we only need the stroke paths and a thickness parameter (typically slope steepness or lighting).

4 Results and discussion

We present results for terrain models rendered by our system, each using a different NPR technique reproducing loose lines and hachures. Running times were gathered from a 2 GHz Pentium IV with OpenGL/Mobility Radeon 7500 (32 MB) graphics.

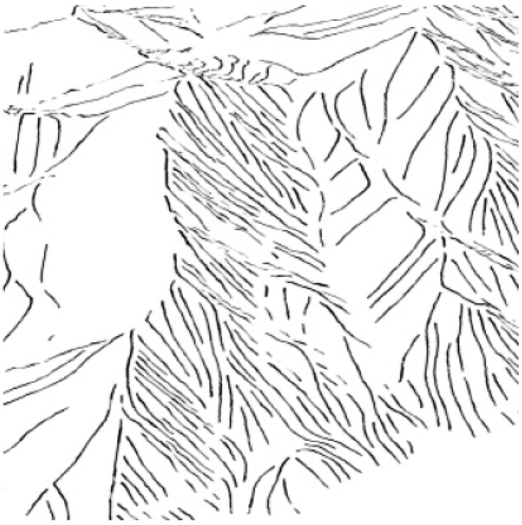


Figure 6: Selecting strokes by light and rendering them as loose lines.

We notice that changing the perturbation, thickness parameter, or the lighting requires a linear amount of time in the number of strokes (more precisely: number of stroke vertices). As demonstrated on the accompanying video file [1], the strokes exhibit temporal coherence when we fly by the models and change light setups. The reason is that we pre-compute all possible stroke paths. During runtime, the only (optional) changes to the rendering are due to new lighting setups and to adjustments to the stroke attributes (subsec. 3.4). Also, the stroke density remains with a good uniform distribution across the model, regardless of the camera and light setups.

[1]http://www.cpsc.ucalgary.ca/~mario/projects/2004/MCW/index.htm

Figures 6, 7 and 8 show results for loose line drawings. We use the terrain data of *Queenstown/Wakatipu Basin, New Zealand*, with a DEM_{size} of 800x800, 8 layers, 5000 stroke paths, with the following number of strokes per layer $(510, 563, 602, 687, 720, 817, 873)$. Timings (in secs.) are as follows: perturbation (0.21), thickness (0.43), lighting (0.01), construction (25.967). Figure 6 demonstrates the effect of loose lines with procedural perturbation and thickness adjustment with a left positioned light for a larger area of the terrain. Figure 7 shows the effect of using procedural and texture-based approaches for stroke qualities. Notice that terrain slope lines are clearly depicted with visual qualities resembling traditional loose line illustrations (fig. 1 (a, b)). Finally in Figure 8 the same terrain model is shown using the stroke tone value map of Table 1. Notice the gradation of ink stroke tones depicting the light intensities.

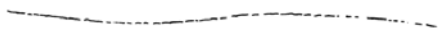
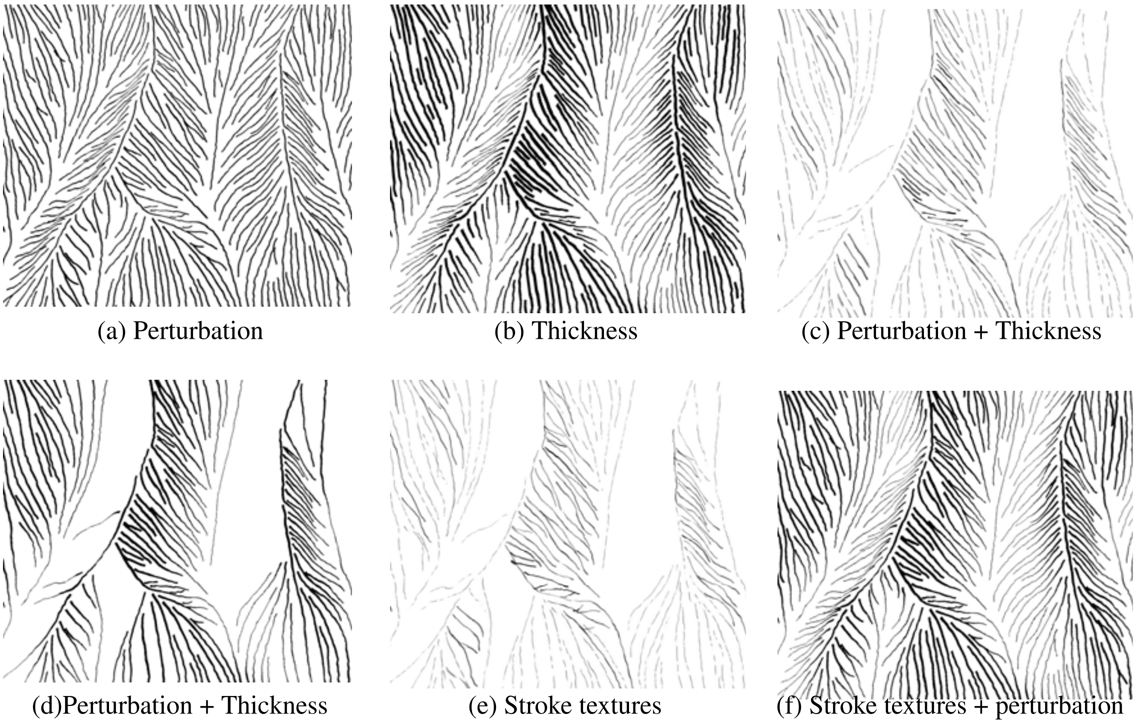
Tone value	Stroke texture
0.0 - 0.25	
0.25 - 0.5	
0.5 - 0.75	
0.75 - 1.0	

Table 1: Texture samples from hand-drawn pen-and-ink strokes mapped to a stroke tone map. Notice that the last tone value range maps to no texture (white)

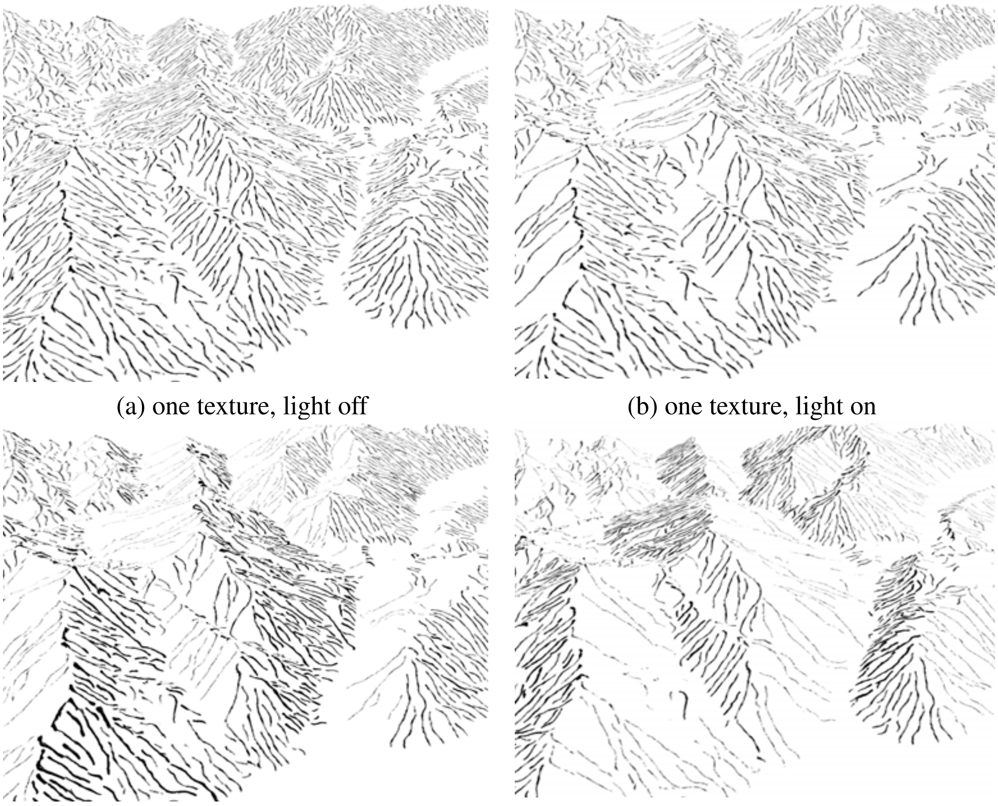
Figures 9, 10 and 11 show results for hachure line drawings. In Figures 9 and 11 we use the model *Uetliberg, Switzerland* with a DEM_{size} of 413x287, 8 layers, 2322 stroke paths, with the following number of strokes per layer $(126, 147, 155, 263, 313, 394, 441, 483)$. Timings (in secs.) are as follows: perturbation (0.08), thickness (0.18), lighting (< 0.01), construction (3.826). The first two rows in Figure 9 reproduce the classical hachuring methods using slope steepness and lighting, respectively, to control stroke thickness. Compare with real hand-drawn hachures for those effects in Figure 1(c). In Figure 10 we demonstrate the effect of our NPR hachure algorithm with MV scale variation applied to a terrain model of *Kawai, Hawaii*. Notice that our algorithm still preserves the hachure style under different visualization results across the four sections of the terrain model. Finally, Figure 11 illustrates the effect of high-contrast for depicting 3D forms, applied to the same terrain model of Figure 9.



(a) Perturbation (b) Thickness (c) Perturbation + Thickness

(d)Perturbation + Thickness (e) Stroke textures (f) Stroke textures + perturbation

Figure 7: Loose line drawing using procedural (a, b, c, d, f) and texture (e, f) approaches. Light is turned off (a, b, c) and on (d, e, f).



(a) one texture, light off (b) one texture, light on

(c)three textures, light on (d) three textures, different light position

Figure 8: Loose lines with texture-based approach. (a, b) using texture sample from Table 1, tone value 0.25-0.5 and (c, d) using three textures from Table 1.

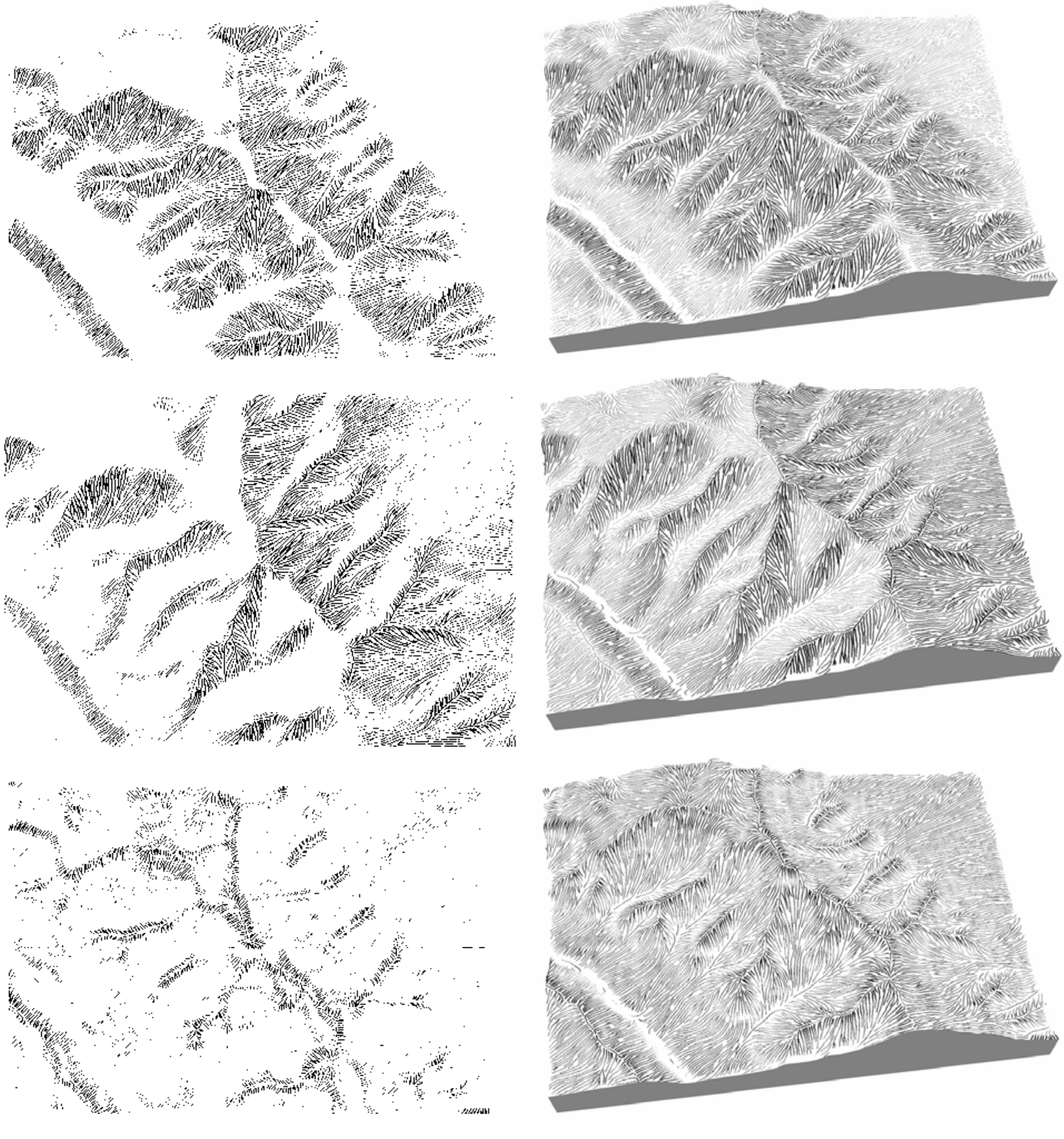


Figure 9: Hachure textures (left column) and projected onto the terrain geometry (right column). The first two rows use slope steepness (top) and lighting (middle) as thickness parameter as in traditional hachure maps, (compare with figure 1c). In contrast, in the bottom row, profile curvature is used as thickness parameter.

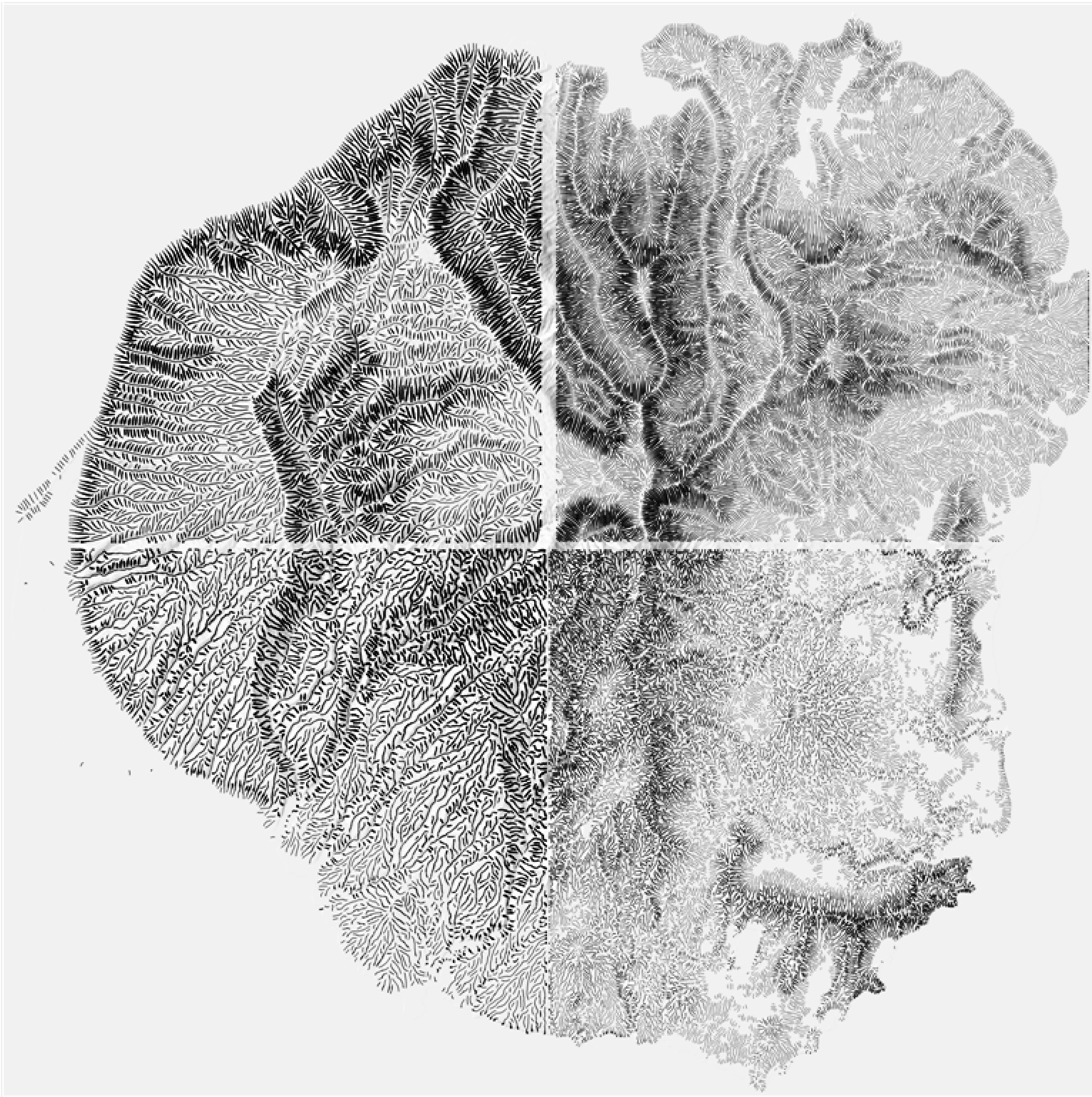


Figure 10: Terrain size: 413x287, total number of strokes: 2322. Top: lighting on, perturbation off, stroke texture off, thickness: profile curvature. Bottom: applying one stroke texture.

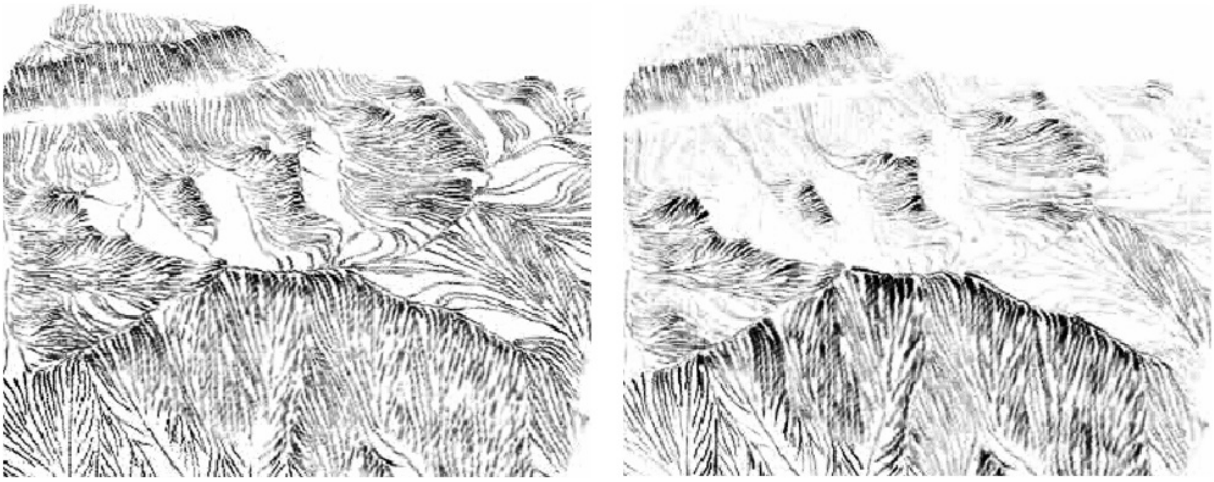


Figure 11: Terrain size: 413x287, total number of strokes: 2322. Left: lighting on, perturbation off, stroke texture off, thickness: profile curvature. Right: applying one stroke texture.

5 Conclusions and future work

We introduced a NPR line drawing pipeline to reproduce traditional terrain slope relief illustration techniques. Line drawings of terrain communicate terrain shape in an image-space efficient way, i.e., strokes do not fully cover the terrain surface. This way, image space remains for visualizing additional thematic information on top of the terrain surface, which represents an important requirement for interactive, explorative geo-data analysis.

The rendering pipeline components are encapsulated from the actual terrain geometry using a texture-mapping approach, raster data (in our case strokes) is projected onto the terrain. This approach integrates well with a wide range of texture-based visualization techniques such as texture-based vector data rendering [11], also taking fully advantage of today's graphics hardware. We described a stroke placement scheme for maintaining frame-to-frame coherence during changes in view and lighting, also allowing rendering at interactive rates. We also demonstrate that the line rendering pipeline can be used to approximate the reproduction of the traditional techniques of loose and hachure lines.

In future work, we would like to address how to improve terrain line drawings based on additional terrain semantics such as explicitly modeled ridges or land-use information, in order to optimize position, direction, and appearance of individual strokes. This would give users additional means of terrain design. Also, our approach considers only view independent feature lines. We would like to investigate the integration of P-strokes and silhouettes, and other stroke based styles of terrain depiction, such as landscape paintings. We also plan to explore different strategies to place the seed points, for instance, according to the importance of the underlying area.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft within the European graduate program 'Combinatorics, Geometry, and Computation' (No. GRK 588/2) and by the Natural Sciences and Engineering Research Council of Canada.

References

[1] K. Baumann, J. Döllner, and K. Hinrichs. Integrated multiresolution geometry and texture models for terrain visualization. In *Proc. of Joint Eurographics - IEEE TVCG Symposium on Visualization 2000*, pages 157–166, 2000.

[2] K. Dowson. *Towards extracting artistic sketches and maps from digital elevation models*. PhD thesis, University of Hull, 1994.

[3] L. Dudley. *Architectural Illustration*. Prentice-Hall, 1977.

[4] I. S. Evans. General geomorphometry, derivatives of altitude, and descriptive statistics. In *Spatial analysis in Geomorphology*, pages 17–90. Methuen and Co. Ltd., 1972.

[5] L. Hurni, E. Hutzler, and A. Neumann. Digital cliff drawing for topographic maps. In *Proc. of the 19th International Cartographic Congress*, volume 1, pages 1045–1052, 1999.

[6] L. Hurni, B. Jenny, T. Dahinden, and E. Hutzler. Interactive analytical shading and cliff drawing: Ad-

vances in digital relief presentation for topographics mountain maps. In *Proc. of the 20th International Cartographic Congress*, 2001.

[7] E. Imhof. *Gelände und Karte*. Eugen Rentsch Verlag, 1968.

[8] E. Imhof. *Cartographic Relief Presentation*. Walter de Gruyter, 1982.

[9] B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. In *Proc. of Visualization in Scientific Computing*, pages 43–55, 1997.

[10] P. Kennelly and A. J. Kimerling. Desktop hachure maps from digital elevation models. *Cartographic Perspectives*, 37:78–81, 2000.

[11] O. Kersting and J. Döllner. Interactive 3D visualization of vector data in GIS. In *Proc. of the 10th ACM international symposium on Advances in geographic information systems*, pages 107–112, 2002.

[12] P. L. Lesage and M. Visvalingam. Towards sketch-based exploration of terrain. *Computers & Graphics*, 26:309–328, 2002.

[13] P. Lindstrom, D. Koller, W. Ribarsky, L. Hodges, N. Faust, and G. Turner. Level of detail rendering of height fields. In *Proc. of SIGGRAPH '96*, pages 109–118, 1996.

[14] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[15] E. Praun, H. Hoppe, M. Webb, and A. Finkelstein. Real-time hatching. In *Proc. of SIGGRAPH '01*, pages 579–584, 2001.

[16] M. P. Salisbury, S. E. Anderson, R. Barzel, and D. H. Salesin. Interactive pen-and-ink illustration. In *Proc. of SIGGRAPH '94*, pages 101–108, 1994.

[17] T. T. Sasada. Drawing natural scenery by computer graphics. *Computer-Aided Design*, 19(4):212–218, May 1987.

[18] Peter A. Shary, Larisa S. Sharaya, and Andrew V. Mitusov. Fundamental quantitative methods of land surface analysis. *Geoderma*, 107:1–32, May 2002.

[19] M. C. Sousa and J. W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. *Computer Graphics Forum (Proc. of Euro-Graphics '99)*, 18(3):195–207, 1999.

[20] M. Visvalingam and K. Dowson. Algorithms for sketching surfaces. *Computers and Graphics*, 22(2-3):269–280, 1998.

[21] M Visvalingam and J. C. Whelan. Occluding contours within artistic sketches of surfaces. In *Proc. of EuroGraphics UK '98*, pages 281–289, 1998.

[22] R. Weibel and M. Heller. Digital terrain modelling. In D.J. Maguire, editor, *Geographical Information Systems: Principles and Applications*, pages 17–90. Longman Scientific & Technical, England, 1991.

[23] J. C. Whelan and M. Visvalingam. Formulated silhouettes for sketching terrain. In *Proc. of the Theory and Practice of Computer Graphics '03*, pages 90–96, 2003.

[24] G. Winkenbach and D. H. David H. Salesin. Computer-generated pen-and-ink illustration. In *Proc. of SIGGRAPH '94*, pages 91–100, 1994.

[25] P. Yoeli. Topographic relief depiction by hachures with computer and plotter. *Cartographic Journal*, 22:111–124, 1985.